

1. Chương 1: Ajax và Ứng dụng

1. [Tổng quan về AJAX](#)

2. [Ứng dụng AJAX xây dựng module Từ điển trực tuyến](#)

2. Chương 2: Wap và Ứng dụng

1. [Xây dựng ứng dụng WAP với IIS và ASP](#)

Tổng quan về AJAX

Vào những năm 90, trên thế giới, khi mà công nghệ “Web động” cho phép các lập trình viên nhanh chóng phát triển các ứng dụng trên nền Web với khả năng tương tác 2 chiều với người sử dụng, nhiều người đã có thể hy vọng rằng đến một lúc nào đó tất cả các ứng dụng mà chúng ta sử dụng sẽ là các ứng dụng Web thay vì các phần mềm chạy độc lập trên các máy tính đơn lẻ (ứng dụng desktop). Quả thật, với sự phát triển chóng mặt của mạng Internet cùng với những ưu điểm của các ứng dụng Web (truy cập tại mọi nơi, không cần nâng cấp nhiều về phía máy sử dụng,...), tương lai của các phần mềm chắc chắn sẽ gắn chặt với các ứng dụng Web, nếu không muốn nói là có thể sẽ bị thay thế. Tuy nhiên, cho đến giờ, hy vọng đó vẫn chưa được đáp ứng một cách triệt để bởi một số điểm đặc trưng của ứng dụng Web lại chính là những giới hạn tưởng chừng như không thể vượt qua nổi. Đó chính là cách thức mà người dùng và ứng dụng Web tương tác với nhau.

Tổng quan về AJAX

Khác với các phần mềm chạy độc lập ở máy khách (có khả năng tương tác gần như tức thời với người dùng), các ứng dụng Web bị giới hạn bởi chính nguyên lý hoạt động của nó: tất cả các giao dịch phải thực hiện thông qua phương thức giao dịch HTTP (HyperText Transport Protocol - Giao thức truyền tải qua các siêu liên kết) trong một mô hình có tên Client/Server. Bất kỳ một tác động nào của người dùng lên ứng dụng Web thông qua trình duyệt đều cần thời gian gửi về Server và sau khi xử lý, Server sẽ trả về những thông tin người dùng mong đợi. Như vậy, độ trễ trong trường hợp này chính là điều mà các ứng dụng Web khó có thể sánh với như các phần mềm chạy trên máy tính đơn lẻ (đặc biệt là trong các ứng dụng như Bản đồ trực tuyến, soạn thảo văn bản trực tuyến, sát hạch trực tuyến có tính thời gian làm bài...).

Thuật ngữ AJAX được xuất hiện vào ngày 18/2/2005 trong một bài báo có tên AJAX : A New Approach to Web Applications [1] của tác giả Jesse James Garrett, công ty AdapativePath. Ông định nghĩa và tóm gọn lại từ cụm từ “Asynchronous JavaScript+CSS+DOM+XMLHttpRequest”. Ngay

sau đó thuật ngữ AJAX được phổ biến cực kỳ nhanh chóng trong cộng đồng phát triển Web và cho đến nay nó là một trong những từ khóa được tìm kiếm nhiều nhất trên Internet.

Nội dung định nghĩa của Garrett về AJAX như sau:

AJAX không phải là một công nghệ. Nó là tập hợp của nhiều công nghệ với thế mạnh của riêng mình để tạo thành một sức mạnh mới. AJAX bao gồm:

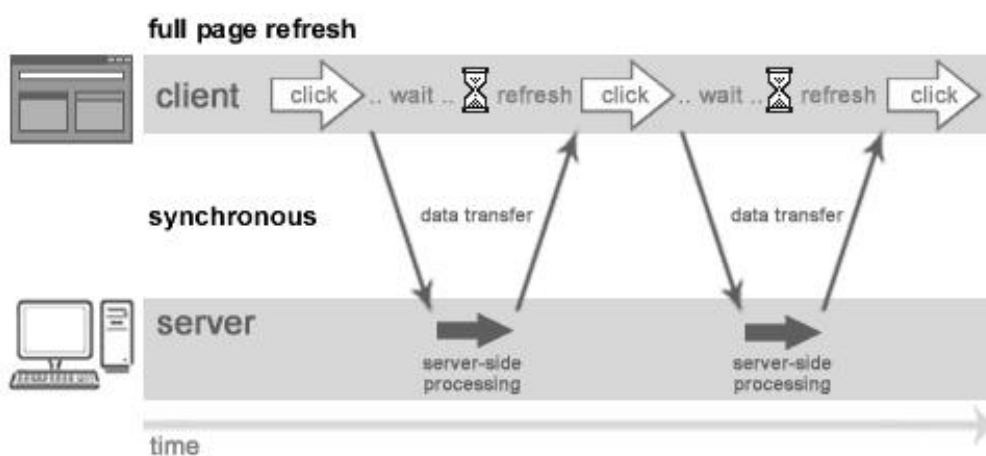
- Thể hiện Web theo tiêu chuẩn XHTML và CSS;
- Nâng cao tính năng động và tương tác bằng DOM (Document Object Model);
- Trao đổi và xử lý dữ liệu bằng XML và XSLT;
- Truy cập dữ liệu theo kiểu bất đồng bộ (asynchronous) bằng XMLHttpRequest;
- Và tất cả các kỹ thuật trên được liên kết lại với nhau bằng JavaScript.[1]

Trong các thành phần cấu thành trên, điểm mấu chốt của AJAX nằm ở XMLHttpRequest. Đây là một kỹ thuật do Microsoft khởi xướng và tích hợp lần đầu tiên vào IE5 dưới dạng một ActiveX. Mozilla tích hợp công nghệ này vào Mozilla 1.0/Netscape 6 sau đó (đương nhiên toàn bộ các version sau này của Firefox đều có XMLHttpRequest) và hiện nay đã có trong trình duyệt Safari 1.2 (Apple) và Opera 7 trở lên. Các vấn đề về XMLHttpRequest và cách sử dụng nó trên các trình duyệt khác nhau sẽ được chúng ta quay trở lại ở các phần tiếp theo. Sau đây là một số nét khác biệt cơ bản giữa các ứng dụng Web truyền thống và ứng dụng Web sử dụng AJAX.

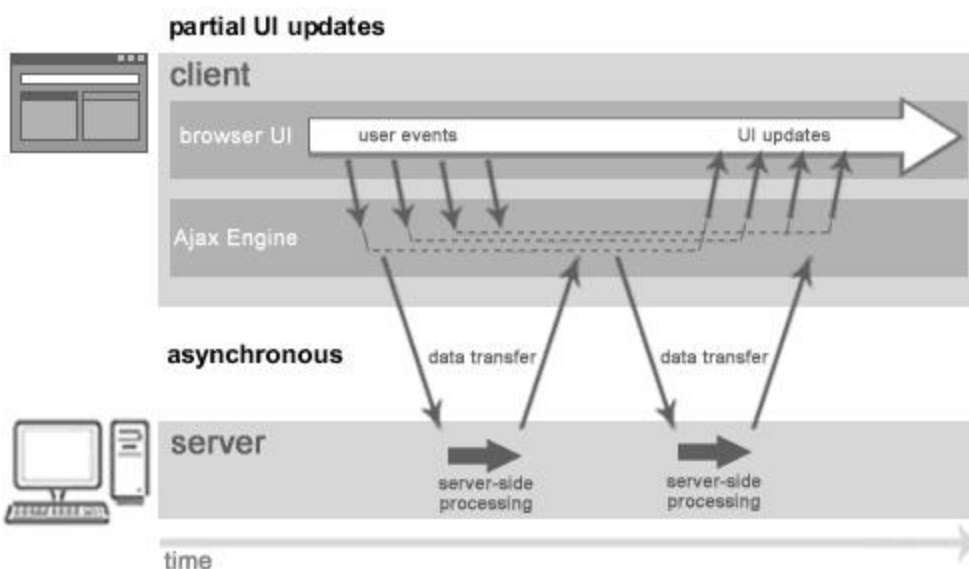
Trong các ứng dụng Web truyền thống, khi người dùng có một cần thay đổi dữ liệu trên trang Web, yêu cầu thay đổi được gửi về server dưới dạng HTTP request (hay còn gọi postback), server sẽ xử lý yêu cầu này và gửi trả lại trang HTML khác thay thế trang cũ. Qui trình này được mô tả là nhấp-chờ và tải lại (click-wait-and-refresh): ví dụ người dùng sau khi nhấn một nút “Submit” trên trang Web phải chờ cho đến khi server xử lý

xong mới có thể tiếp tục công việc. Ngược lại, trong các ứng dụng AJAX, người dùng có thể nhấn chuột, gõ phím liên tục mà không cần chờ đợi. Nội dung tương ứng với từng hành động của người dùng sẽ gần như ngay lập tức được hiển thị vào vị trí cần thiết (đáp ứng gần như tức thời) trong khi trang Web không cần phải refresh lại toàn bộ nội dung. Để tìm hiểu kỹ hơn điều này, ta sẽ xem xét 2 mô hình ứng dụng như [1] đã đề cập, Mô hình cổ điển và Mô hình AJAX-based:

Mô hình cổ điển của một ứng dụng Web (hình minh họa được lấy từ bài báo của Garrett)



Mô hình ứng dụng Web sử dụng AJAX (theo bài báo của Garrett)



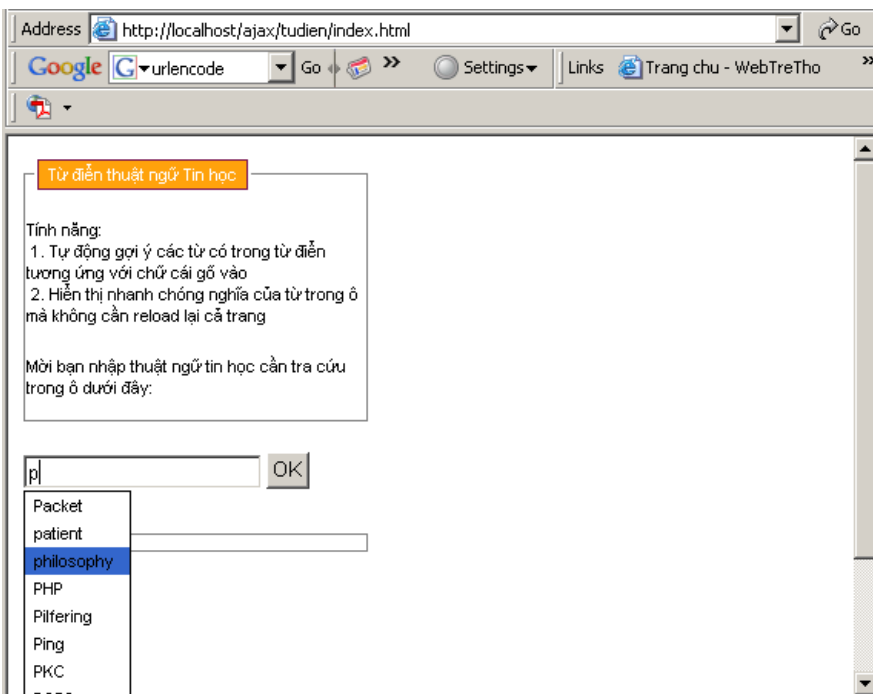
Rõ ràng điểm khác biệt là thay vì phải tải cả trang Web thì với AJAX trình duyệt phía người dùng chỉ cần tải về phần của trang Web mà người dùng muốn thay đổi. Điều này giúp cho ứng dụng Web phản hồi nhanh hơn, thông minh hơn. Ngoài ra, điểm đặc biệt quan trọng trong công nghệ AJAX nằm ở chữ A (Asynchronous) – không đồng bộ – tức là người dùng cứ gửi yêu cầu của mình tới server và quay lại với công việc của mình mà không cần chờ trả lời. Khi nào server xử lý xong yêu cầu của phía người dùng, nó sẽ báo hiệu và người dùng có thể “thu nhận lấy” để thể hiện những thay đổi cần thiết. Vậy tất cả cơ chế này hoạt động thực sự thế nào? AJAX cho phép tạo ra một AJAX Engine nằm giữa giao tiếp này. Khi đó, các yêu cầu gửi (request) và nhận (response) do AJAX Engine thực hiện. Thay vì trả dữ liệu dưới dạng HTML và CSS trực tiếp cho trình duyệt, Web server có thể gửi trả dữ liệu dạng XML và AJAX Engine sẽ tiếp nhận, phân tách và chuyển hóa thành XHTML + CSS cho trình duyệt hiển thị. Việc này được thực hiện trên client nên giảm tải rất nhiều cho server, đồng thời người sử dụng cảm thấy kết quả xử lý được hiển thị tức thì mà không cần nạp lại trang. Mặt khác, sự kết hợp của các công nghệ Web như CSS và XHTML làm cho việc trình bày giao diện trang Web tốt hơn nhiều và giảm đáng kể dung lượng trang phải nạp. Đây là những lợi ích hết sức thiết thực mà AJAX đem lại. Chúng ta sẽ xem xét cụ thể các thành phần cấu thành AJAX, nguyên lý hoạt động và việc sử dụng Javascript kết nối chúng trong phần tiếp theo.

Ứng dụng AJAX xây dựng module Từ điển trực tuyến

Sau khi nắm được nguyên lý của AJAX, bạn đọc có thể sử dụng một công cụ lập trình web bất kỳ để xây dựng các ứng dụng tiện ích như tra cứu từ điển (giống google suggest), tự động sao lưu văn bản đang soạn (auto save), tự động sao lưu các giá trị form đang nhập...

Mô tả Ứng dụng

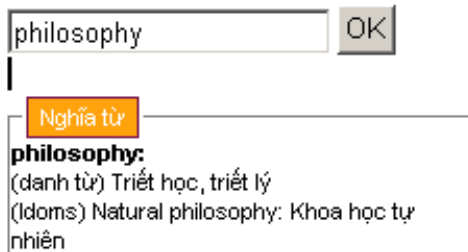
Dưới đây là màn hình module phần mềm từ điển trực tuyến sử dụng công nghệ AJAX [Lien ket](#)



vidu

Người dùng gõ vào một từ cần tra cứu bất kỳ, sau mỗi ký tự được nhập vào ô tra cứu, danh sách các từ trong CSDL từ điển có các ký tự tương ứng ngay lập tức hiện ra trong một danh sách đổ xuống, giúp người dùng dễ dàng nhập liệu và lựa chọn (giống cách làm việc của Google Suggest)

Sau khi có từ cần tra cứu, người dùng nhấn ok, nghĩa của từ đó ngay lập tức hiện ra phía dưới (không cần reload lại trang)



hình minh họa 2

Phía người dùng chỉ biết đến 1 trang duy nhất là index.html. Hậu trường của ứng dụng là cơ chế Ajax được thực hiện trong các file javascript included và các file asp hỗ trợ truy xuất CSDL qua các tham số gửi theo đường XMLHttpRequest từ file ajax.js và ajax_search.js.

Mã nguồn minh họa

Đoạn code minh họa đơn giản dưới đây (viết trên ASP và CSDL Access) sẽ giúp bạn hiểu rõ hơn cơ chế này.

Tệp index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html lang="en-US">
```

```
<head>
```

```
<style type="text/css" media="screen">
```

```
body {
```

```
font: 11px arial;}
```

```
.suggest_link {
```

```
background-color: #FFFFFF;
```

```
padding: 2px 6px 2px 6px;}
```

```
.suggest_link_over {
```

```
background-color: #3366CC;
```

```
padding: 2px 6px 2px 6px;}
```

```
#search_suggest {
```

```
position: absolute;
```

```
background-color: #FFFFFF;
```

```
text-align: left;
```

```
border: 1px solid #000000;}
```

```
label {
```

```
width:4em;
```

```
float:left
```

```
text-align:right;
```

```
margin-right:0.5em;
```

```
display:block;}
```



```
.submit input{
margin-left:4.5em;}

fieldset{
border:1px solid #781351;
width:20em;}
```

```
Legend {
color:#fff;
background:#ffa20c;
border:1px solid #781351;
padding:2px 6px}
```

```
</style>
```

```
<script language="JavaScript" type="text/javascript"
```

```
src="ajax_search.js"></script>
```

```
<script language="JavaScript" type="text/javascript" src="ajax.js">
</script>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
```

```
</head>
```

```
<body>
```

```
<fieldset><legend>Từ điển thuật ngữ Tin học</legend>
```

```
<p>Tính năng: <br>&nbsp;1. Tự động gợi ý các từ có trong từ điển tương
ứng với chữ cái gõ vào<br>&nbsp;2. Hiện thị nhanh chóng nghĩa của từ
```

trong ô mà không cần reload lại cả trang </p>

<p>Mời bạn nhập thuật ngữ tin học cần tra cứu trong ô dưới đây:

 </p>

</fieldset>

<form name=frmSearch id="frmSearch" method=get>

<input type="text" id="formWord" name="formWord" alt="Search
Criteria" onkeyup="searchSuggest();" autocomplete="off" />

<input type="button" value="OK"

onclick="sndUserCheck(document.frmSearch.formWord.value);"> </td>

<div id="search_suggest"></div>

</form>

<p></p>

<fieldset style="width: 220; height: 39">

<legend>Nghĩa từ</legend>

<div id="TakenOrNot"></div>

</fieldset>

</body>

</html>

Tệp ajax.js

```
function createRequestObject()

{

var ro;

var browser = navigator.appName;

if(browser == "Microsoft Internet Explorer") {

// on IE, we have to use ActiveX

ro = new ActiveXObject("Microsoft.XMLHTTP");

} else {

// on every other browser, we can directly create a new XMLHttpRequest
object

ro = new XMLHttpRequest();

}

return ro;

}

var http = createRequestObject();

// this function should be called for user input

// the response in this case is formatted as follows:

// object|text

// where object is the id of the HTML element we are going to update

// and text is what it will be updated to

// this could obviously work a lot better with some XML
```

```
function handleResponse()
{
    if(http.readyState == 4) {
        var response = http.responseText;
        var update = new Array();
        if(response.indexOf('|' != -1)) {
            update = response.split("|");
            document.getElementById(update[0]).innerHTML = update[1];
        }
    }
}

// this function should be called for user input

// it opens up the usercheck.asp page with a querystring of 'action'
function sndUserCheck(action)
{
    http.open("get", "searchMeaning.asp?formWord=" + action);
    http.onreadystatechange = handleResponse;
    http.send(null);
}
```

Tệp ajax_search.js

```
//Gets the browser specific XmlHttpRequest Object

function getXmlHttpRequestObject() {

    if (window.XMLHttpRequest) {

        return new XMLHttpRequest();

    } else if(window.ActiveXObject) {

        return new ActiveXObject("Microsoft.XMLHTTP");

    } else {

        alert("Your Browser Sucks!\nIt's about time to upgrade don't you think?");

    }

}

//Our XmlHttpRequest object to get the auto suggest

var searchReq = getXmlHttpRequestObject();

//Called from keyup on the search textbox.

//Starts the AJAX request.

function searchSuggest() {

    if (searchReq.readyState == 4 || searchReq.readyState == 0) {

        var str = escape(document.getElementById('formWord').value);

        searchReq.open("GET", 'searchSuggest.asp?formWord=' + str, true);

        searchReq.onreadystatechange = handleSearchSuggest;
```

```
searchReq.send(null);

}

}

//Called when the AJAX response is returned.

function handleSearchSuggest() {

if (searchReq.readyState == 4) {

var ss = document.getElementById('search_suggest')

ss.innerHTML = "";

var str = searchReq.responseText.split("<br>");

for(i=0; i < str.length - 1; i++) {

//Build our element string. This is cleaner using the DOM, but

//IE doesn't support dynamically added attributes.

var suggest = '<div onmouseover="javascript:suggestOver(this);" ';

suggest += 'onmouseout="javascript:suggestOut(this);" ';

suggest += 'onclick="javascript:setSearch(this.innerHTML);" ';

suggest += 'class="suggest_link">' + str[i] + '</div>';

ss.innerHTML += suggest;

}

}

}
```

```
//Mouse over function

function suggestOver(div_value) {

div_value.className = 'suggest_link_over';

}

//Mouse out function

function suggestOut(div_value) {

div_value.className = 'suggest_link';

}

//Click function

function setSearch(value) {

document.getElementById('formWord').value = value;

document.getElementById('search_suggest').innerHTML = "";

}
```

Tệp search_suggest.asp

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="65001"%>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Module tu dien</title>

</head>
```

```
<%
```

```
//Written by Minhden
```

```
//Lay bien search bang request, sau do search trong CSDL
```

```
//nhung dong nao co title like bien search thi lay ra
```

```
%>
```

```
<!--#include file="connection.asp"-->
```

```
<%///Make sure that a value was sent.
```

```
formWord=request("formWord")
```

```
//Get every page title for the site.
```

```
if len(formWord)>=1 then
```

```
str = "SELECT * FROM dictionary WHERE word like '"&formWord&"%'
ORDER BY word"
```

```
rs.open str,conn
```

```
do until rs.eof%>
```

```
<%=rs("word")%><br>
```

```
<%rs.movenext
```

```
loop
```

```
rs.close
```

```
set rs=nothing
```

```
end if
```

```
%>
```


Tệp Search_mearning.asp

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="65001"%>

dim strConn, dbConn, RS

set dbconn= server.createObject("ADODB.connection")

dbConn.Open "PROVIDER=MSDASQL;DRIVER={Microsoft Access
Driver (*.mdb)};DBQ=" & Server.MapPath("dictionary.mdb")

set rs = server.createObject("ADODB.recordset")

'query the database

str="SELECT * FROM dictionary WHERE word = " &

request.querystring("formWord")&""

set RS = dbConn.execute(str)

'is this name taken or not

if not RS.eof then

response.write "TakenOrNot|<b>"&request.querystring("formWord") &":
</b><br>"&rs("meaning")

else

response.write "TakenOrNot|Thong bao: " & "Khong co tu nay"

'end if

end if

%>
```

Tương tự như vậy, các ứng dụng kiểm tra tên đăng nhập (người dùng chọn tên đăng nhập và check xem tên đó đã có chưa), tự động sao lưu văn bản đang soạn thảo (auto save), tự động sao lưu nội dung làm việc của người dùng (ví dụ điền form)... cũng được thực hiện thông qua cơ chế này.

Chúc các bạn thành công

Xây dựng ứng dụng WAP với IIS và ASP

WAP, viết tắt của Wireless Application Protocol (Giao thức ứng dụng không dây), kế thừa các chuẩn Internet (HTML, XML và TCP/IP), cho phép các thiết bị cầm tay có thể kết nối tới Server truy xuất thông tin và các dịch vụ. Giao thức này được thiết kế cho các trình duyệt siêu nhỏ (micro browser) dùng ngôn ngữ đánh dấu WML (Wireless Markup Language). Công nghệ WAP đang là công nghệ chuẩn chủ đạo cho các ứng dụng trên các thiết bị không dây như điện thoại di động. Một số ứng dụng WAP điển hình đang được áp dụng: đặt vé; kiểm tra email; xem dự báo thời tiết, tỉ giá, giá cổ phiếu; xem kết quả bóng đá; tra cứu danh bạ điện thoại...

Xây dựng ứng dụng WAP với IIS và ASP

Đỗ Ngọc Minh

Viện CNTT, ĐH Quốc Gia Hà Nội

Email: minhhdn@vnu.edu.vn

1. Giới thiệu

WAP là viết tắt của Wireless Application Protocol (Giao thức ứng dụng không dây), được kế thừa các chuẩn Internet (HTML, XML và TCP/IP), cho phép các thiết bị cầm tay có thể kết nối tới Server truy xuất thông tin và các dịch vụ. Giao thức này được thiết kế cho các trình duyệt siêu nhỏ (micro browser) nhờ sử dụng ngôn ngữ đánh dấu WML (Wireless Markup Language). WML được định nghĩa như là ứng dụng XML 1.0

Công nghệ Wap ra đời vào năm 1997 bởi Wap forum (www.wapforum.org), bao gồm Ericsson, Motorola, Nokia, và Unwired Planet. Công nghệ Wap đang là công nghệ chuẩn chủ đạo cho các ứng dụng phục vụ trên các thiết bị không dây như các điện thoại di động kỹ thuật số. Phiên bản mới nhất gần đây là phiên bản Wap 2.0.

Các nhà cung cấp dịch vụ Internet hiện nay không những chú trọng phát triển, cập nhật thông tin cho Website của họ mà còn rất chú trọng tới các

thông tin dành cho WAP. Với sự ra đời của ngày càng nhiều các loại điện thoại có khả năng hỗ trợ Wap cùng với sự phát triển của hệ thống thông tin di động, chắc chắn Wap sẽ có một vai trò quan trọng trong xã hội hiện đại.

Một số ứng dụng Wap điển hình đang được áp dụng:

- Kiểm tra thông tin về chỗ trên tàu.
- Đặt mua vé
- Kiểm tra chuyến bay
- Xem dự báo thời tiết
- Xem giá cổ phiếu
- Tra cứu danh bạ điện thoại
- Check mail
- Xem kết quả bóng đá...

Các trình duyệt Wap siêu nhỏ (WAP Micro Browsers)

Để vừa vận hiển thị các trang thông tin lên các thiết bị không dây cỡ nhỏ, Wap sử dụng trình duyệt siêu nhỏ.

Một trình duyệt siêu nhỏ thực chất là một phần mềm nhỏ tốn rất ít tài nguyên bộ nhớ cũng như các yêu cầu về phần cứng và CPU. Nó có thể hiển thị thông tin nhờ ngôn ngữ đánh dấu gọi là WML

Trình duyệt siêu nhỏ cũng có khả năng thông dịch các phiên bản thu nhỏ của Java Script, gọi là WML Script.

Một số các trình duyệt Wap thông dụng hiện nay như: Klondike, Deck-it, Gelon... Có thể download các phần mềm trình duyệt này về để thực thi và hiển thị các trang wml trên máy tính của bạn từ các địa chỉ sau:

Deck-it: <http://www.Pyweb.com>

Klondike: <http://www.apachesoftware.com> ...

Gelon: <http://www.gelon.net>

WML là gì?

WML là viết tắt của Wireless Markup Language. Nó là ngôn ngữ đánh dấu kế thừa từ HTML, tuy nhiên WML dựa trên XML, do đó nó chặt chẽ hơn HTML.

WML được sử dụng để tạo ra các trang có khả năng hiển thị trên WAP browser. Các trang trên WML gọi là DECKS. Decks được thiết lập bởi tập hợp các CARDS.

Ví dụ một trang WML :

```
<?xml version="1.0"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
```

```
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<card id="card1" title="Minh hoa card 1">
```

```
<p>
```

Chao ban da den voi wml

```
</p>
```

```
</card>
```

```
</wml>
```

Bạn có thể tham khảo thêm về cú pháp của WML tại địa chỉ www.w3schools.com/wap .

Trang wml này sau khi save vào đĩa và hiển thị nhờ Klondike (Trình duyệt này cho phép mở trực tiếp trang wml bình thường như mở 1 file)	Hiển thị nhờ Deck-it:
--	-----------------------

Một số so sánh minh họa sự khác biệt giữa HTML và WML

HTML	WML
- Các trang có đuôi .htm hoặc .html- Hiển thị nhờ trình duyệt Web (Web browser)IE, Netscape, Opera ...	- Các trang có đuôi .wml- Hiển thị nhờ trình duyệt Wap (Wap browser)Klondike, Deck-it, Gelon...
Cú pháp không chặt chẽ	Cú pháp rất chặt chẽ do sử dụng công nghệ XML
Một trang html được gọi là 1 Web page	Một trang wml được gọi là một card
Sử dụng Java Script để chạy các ứng dụng đơn giản trên ClientMã Java Script có thể được trộn lẫn trong mã HTML	Sử dụng WML Script (chính là Java Script thu gọn)Mã WML Script phải nằm ngoài trang WML.

Cho phép hiển thị thông tin cực kỳ đa dạng, phong phú (Các hiệu ứng âm thanh hình ảnh...)	Bị giới hạn bởi màn hình thiết bị (thường là rất bé), nếu có hiển thị ảnh thì cũng phải sử dụng format riêng, ví dụ WBMP có kích thước và dung lượng file rất nhỏ, ngoài ra, đây là ảnh đen trắng.
---	--

2. Cấu hình phần mềm máy chủ IIS (Internet Information Services) cho phép thực thi chức năng của Wap Server

Như ta đã biết, từ Windows 2000, IIS được tích hợp sẵn trong bộ cài và là một thành phần cài đặt tùy chọn (option). Các cấu hình mặc định trong IIS cho phép IIS là 1 server phục vụ các ứng dụng Web, tuy nhiên muốn IIS của ta có thêm khả năng phục vụ các ứng dụng Wap, cần thêm một số thông số như sau:

1. Từ menu Start, chọn Setting - Control Panel – Administrative tools - Internet Services Manager. Mở rộng dấu + ở tên máy và click chuột phải vào default web site
2. Lựa chọn properties.
3. Chọn Tab HTTP headers, ở mục MIME map, chọn file types
4. Click New Type.
5. Gõ wml vào ô Extension
6. Gõ text/vnd.wap.wml vào ô Content MIME.
7. Lặp lại các bước từ 4 đến 6 và gõ vào các thông số theo bảng sau

Bảng 1: MIME Types and File Extensions	
Type	File Extension
text/vnd.wap.wml	.wml

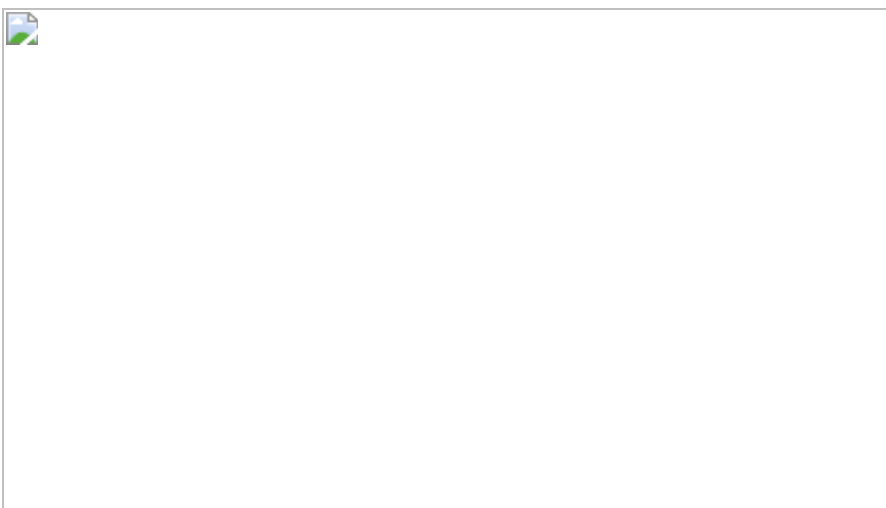
application/vnd.wap.wmlc	.wmlc
text/vnd.wap.wmlscript	.wmls
application/vnd.wap.wmlscriptc	.wmlsc
Image/vnd.wap.wbmp	.wbmp

Để xây dựng các ứng dụng WML cho phép tương tác với người sử dụng giống như công việc xây dựng trang Web động ta cũng lựa chọn công nghệ ASP (Active Server Pages) cho phép thực thi các ứng dụng trên máy chủ trước khi trả lời Client. Nếu bạn đã từng sử dụng ASP cho các ứng dụng Web thì bạn sẽ thấy một điều cơ bản là ASP sau khi thực thi trên máy chủ sẽ trả về cho Client mã HTML là mã mà trình duyệt có thể hiểu được. Như vậy, khi chuyển sang xây dựng ứng dụng trên WML, để hiển thị được trang trả về trên wap browser thì bạn chỉ cần chỉ định ASP trả về mã WML là xong. Các thao tác còn lại giống hệt như truy xuất Web động.

Hiện tại WML của ta vẫn ở trạng thái tĩnh, ta cần phải thêm vào asp.dll để IIS có khả năng tạo ra văn bản wml. Để làm điều này, trong mục Properties của Default webserver, chọn tab Home Directory, Click Configuration, bảng sau hiện ra:



ta chọn add, sau đó điền các nội dung vào bảng Add/Edit Application Extension Mapping như sau:



Như vậy là ta đã cấu hình xong IIS, ngoài ra ở trên đầu mỗi trang asp, muốn hiển thị trên wap browser ta cần thêm đoạn lệnh sau:

```
<% response.ContentType = "text/vnd.wap.wml" %>
```

Sau khi cấu hình máy chủ, ta có thể tạo riêng 1 thư mục trên ổ C, ví dụ: C:\WAP-applications, sau đó chép file wml ở trên đặt tên file là index.wml. Ngoài ra, ta gõ thêm file sau và đặt tên là index.asp

```
<% response.ContentType = "text/vnd.wap.wml" %>
```

```
<?xml version="1.0"?>
```

```
<wml>
```

```
<card id="card1" title="Minh hoa asp">
```

```
<p>
```

```
<%response.write("Chao ban")%>
```

```
</p>
```

```
</card>
```

```
</wml>
```

Thiết đặt đường dẫn ảo (Virtual directory) tới thư mục này mang tên wap, sau đó khởi động trình duyệt Wap, gõ vào ô địa chỉ dòng chữ: <http://localhost/wap/index.wml> và <http://localhost/wap/index.asp> Nếu cả hai trường hợp trên màn hình trình duyệt hiện ra nội dung Chao ban có nghĩa là máy chủ của ta đã sẵn sàng để chạy các ứng dụng tiếp theo.

3. Minh họa ứng dụng hiển thị và cập nhật CSDL qua Wap Browser

3.1. Chúng ta cùng nhau xây dựng một ứng dụng nhỏ cho phép hiển thị CSDL là 1 danh sách bao gồm tên và số điện thoại tương ứng của người

đó. CSDL của ta được tạo trên Access, có tên là danhba.mdb, gồm 1 bảng danh sách và 2 trường ten (kiểu text) và dienthoai(kiểu số) như sau:

Table: danhbach	
Ten	Text
Dienthoai	Number

Các file ta sẽ xây dựng tương ứng gồm:

Display.asp cho phép hiển thị CSDL

Form.wml cho phép nhập dữ liệu

Insert.asp cho phép cập nhật dữ liệu từ form vào CSDL

3.2. Mã nguồn từng file tương ứng:

Display.asp

```
<% Response.ContentType = "text/vnd.wap.wml" %>
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"  
"http://www.wapforum.org/DTD/wml_1.1.xml">
```

```
<wml>
```

```
<!-- MAIN CARD-->
```

```
<card id="Danhbadt">
```

<p align="left"><small>Danh ba DT </small></p>

<%

```
strconn = "DRIVER=Microsoft Access Driver (*.mdb);DBQ=" &  
Server.MapPath("danhba.mdb")
```

```
set conn = server.createobject("ADODB.Connection")
```

```
conn.open strconn
```

```
set rs = server.createobject("adodb.recordset")
```

```
Query = "Select * from danhsach"
```

```
rs.open Query, conn
```

```
if not rs.eof Then rs.movefirst
```

```
Do While NOT Rs.EOF
```

```
%>
```

<p align="left">

<small>

<h2><%=rs("ten")%></h2>

<%=rs("dienthoai")%>

</small>

</p>

<%

```
rs.movenext
```

```
Loop
```

else

```
response.write("<p align='left'><small>D.sach chua co ai</small></p>")
```

End if

rs.close

Set conv = nothing

set rs= nothing

set conn = nothing

%>

</card>

</wml>

Form.wml

<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"

"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card title="Input">

<p>

Ten: <input type="text" name="ten" size="15"/>

Dienthoai: <input type="text" name="dienthoai" size="15"/>

</p>

```
<do type="accept" label="Send">  
  
<go href="insert.asp" method="post">  
  
<postfield name="ten" value="$ten"/>  
  
<postfield name="coquan" value="$coquan"/>  
  
<postfield name="dienthoai" value="$dienthoai"/>  
  
</go>  
  
</do>  
  
</card>  
  
</wml>
```

Insert.asp

```
<%
```

```
Dim adoCon
```

```
Dim rsAddnew
```

```
Dim strSQL
```

```
set rs = server.createObject("adodb.recordset")
```

```
Set adoCon = Server.CreateObject("ADODB.Connection")
```

```
adoCon.Open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &  
Server.MapPath("danhba.mdb")
```

```
Set rsAddnew = Server.CreateObject("ADODB.Recordset")
```

```
strSQL = "SELECT * FROM danhsach;"
```

```
rsAddnew.CursorType = 2
```

```
rsAddnew.LockType = 3  
  
rsAddnew.Open strSQL, adoCon  
  
rsAddnew.AddNew  
  
rsAddnew.Fields("ten") = Request("ten")  
  
rsAddnew.Fields("dienthoai") = Request("dienthoai")  
  
rsAddnew.Update  
  
rsAddnew.Close  
  
Set rsAddnew = Nothing  
  
Set adoCon = Nothing  
  
Response.Redirect "danhbadiienthoai.asp"  
  
%>  
  
</p>  
  
</card>  
  
</wml>
```

Ghi cả 3 tập tin trên vào thư mục bạn vừa tạo, khởi động Wap browser rồi gõ vào từ trình duyệt <http://localhost/wap/form.wml> Sau khi bạn nhập tên và số điện thoại rồi submit, nếu màn hình hiển thị tên và số điện thoại bạn vừa nhập thì có nghĩa là bạn đã thành công.

Tương tự như vậy, bạn có thể tham khảo thêm cú pháp các thẻ của WML cũng như WML Script để xây dựng cho mình các ứng dụng phong phú hơn. Riêng đối với ứng dụng CSDL bạn vừa làm, bạn có thể thêm chức năng Xóa, sửa và tìm kiếm để chương trình trên trở thành một ứng dụng hoàn chỉnh.

Chúc các bạn thành công

Tài liệu tham khảo:

1. <http://www.w3schools.com/wap>